MASSACHVSETTS INSTITVTE OF TECHNOLOGY
Department of Electrical Engineering and Computer Science
6.001—Structure and Interpretation of Computer Programs
Spring Semester, 1996

Problem Set 9a

Issued: April 11, 1996

REQUIRED discussion activity and brief writeup due in recitation on FRIDAY, 19 April for *all* sections.
Full writeup due, with problem set 9, on Wednesday, April 24, 1996 for EVEN numbered sections, Friday, April 26, 1996 for ODD numbered sections.

Reading:

- first page of the RSA patent (attached)
- "Should Program Algorithms be Patented?" by Pamela Samuelson (attached)
- "Why the Look and Feel of Software User Interfaces Should Not Be Protected by Copyright Law" by Pamela Samuelson (attached)

This problem set is very different, both in form and in content, from the previous things we have asked you to do this semester. There is no programming assignment. However, there is a **required** activity scheduled for recitation on Friday, 19 April. This is a "role-playing" discussion concerning software property rights. Discussion groups and roles will be assigned in recitation on Friday, April 12. You must get your assigned roles and prepare a brief written plan, which you will turn in on Friday, April 19, before the discussion. After the discussion you will need to write a short summary to turn in with your solution to problem set 9 (April 24 or 26). You *must* attend the Friday, April 19 recitation in order to get any credit for this problem set.

Part 1: Software and intellectual property

You might think that a beginning Computer Science subject like 6.001 should focus only on purely "technical" issues of computers, systems, and languages. But if you go on at all to work in this field, you will also need to confront situations where new technology raises social and ethical issues; and in teaching 6.001 we would like help prepare you for this. Often these situations can be extremely "sticky" both legally and ethically—new technology raises issues at such a rate that our social and legal framework has not yet developed the laws and precedents that would provide agreed upon patterns of conduct. There is rarely one "right way" to resolve such a sticky situation, and reaching a resolution can require patience, creativity, and moral courage.

The intellectual-property status of software—to what extent software technology, programs, and algorithms can be covered by patents and copyrights—has been an increasing source of disputes over the past 10 years, and indications are that things are getting worse. These issues will directly affect you if you go on to be a programmer. For example, you may have read about Lotus Development Corporation's successful "look and feel" copyright suit in which it forced two other spreadsheet products off the market. A key developer for one of these products was a recent MIT undergraduate. Lotus has also sued Borland on similar grounds for "Quattro" and that suit is currently under appeal. You may also have heard about Apple's suit against Hewlett-Packard and Microsoft, in which they claimed that Windows violates Apple copyrights on the "look and feel" of the Macintosh. That claim was largely rejected.[1] You may also have heard about MIT's current disagreement with AT&T over whether the X Window System infringes an AT&T patent. And this problem set asks you to consider a hypothetical but possible situation in which you yourself could be involved in a patent dispute.

In lecture on April 18, you will hear Randy Davis present an overview of the laws concerning the intellectual-property status of software. Here are some of the main points:

- Copyright law and patent law are rather different, in a distinction that goes back to the Constitution: "The Congress shall have the power ... To promote the Progress of Science and the useful Arts, by securing for limited Times to authors and inventors the exclusive Right to their respective Writings and Discoveries."

  Copyrights and patents were not intended by the authors of the Constitution to be entitlements: they don't exist for the sake of the owners of the copyrights or patents. They are not intended as rewards. In fact they are intended to provide incentives for certain behavior with the idea that the modification will benefit the public. The restrictions on use of patented or copyrighted material are thus a price that the public pays in order to gain a benefit that outweighs the cost. Does it? We must consider how much benefit does the public really gets.

  Copyright law was designed to cover the artistic expression of an individual ("Writings"), while patent law covers specific inventions ("Discoveries"). Copyright lasts for the life of the creator plus 50 years (75 years if a corporation holds the copyright), while patents last for 17 years.

  Copyright covers the expression but not the underlying idea, while patents cover the reduction to practice of an idea. Note that in neither case is an "idea" coverable.

  In general, patent coverage is much more restrictive than copyright coverage. For example, you can infringe a patent if you make an invention that is "the same as" the patented one, even if you had never seen or heard of the original. But you can infringe a copyright only if you actually had access to the original and produced something substantially similar.

- The basis for the laws concerning the intellectual-property status of software is extremely complex. Software is covered by both patents and copyrights (as well as by laws covering trade secrets), often with no clear way to determine which form should apply in any particular case.

- Programs have been covered by copyright since the law was updated in 1976. It was widely believed until 1981 that software could not be patented, but an important case in that year opened the door to patenting software.

[1] Incidentally, the clerk for the judge in that case was an MIT student who got her Master's Degree in EECS in 1989 and then went on to attend law school.

- Some recent copyright cases (Whelan, 1986; Lotus, 1990) have extended copyright beyond the literal code to include some of the higher level abstractions (like overall system structure and the user interface). One recent case (Computer Associates, 1991) pointed out fundamental technical difficulties in the Whelan case analysis.
- Software patents were granted at the rate of about 30 a year during the 1970's, about 200 per year during the early 1980's, and over 1000 per year recently. Some of these are filed primarily for defensive purposes.
- There is widespread, intense disagreement over what to do about the current situation. Some people think things are OK: software is just like any other new technology, and if we just wait long enough, there will be sufficient legal precedents to establish a stable framework. Other people think that software—being both an expression of an idea and a mechanism for implementing the idea—is different enough from other forms of inventions that Congress should change the laws covering software, for example, to vastly limit, or even prohibit the possibility of granting patents for software and algorithms. Still others seek to find some way of providing shorter term coverage for inventors, in order to encourage invention and the creation of new software companies, while allowing for the continuing spread of new concepts. The US Patent & Trademark Office just completed hearings on software patents last month and received a wide variety of disagreeing opinions.

The attached papers by legal scholar Pamela Samuelson describe some of the complexities in the debates over patent and copyright coverage of software.

Part 2: MIT's patent on the RSA cryptography system

One of the most famous software patents was issued to MIT in 1983. This is a patent on the RSA method of "public-key cryptography." Public key cryptography is way of encoding messages that does not require the people communicating to exchange any secret encryption and decryption keys. To put it another way, a public-key encryption algorithm is one in which *everyone* can know the precise algorithm used to encrypt a message, and still not be able to decrypt messages that were created using that algorithm. This idea may seem counterintuitive, but it really works. There are public-key encryption methods that so far as anyone knows, are completely unbreakable.

This makes public-key encryption particularly appropriate for use on computer networks. For example, you can "publish" a method that people can use in order to encrypt the E-mail that they send to you. Even though everyone will know the method and be able to send you encrypted messages, only you will be able to decrypt the messages. A similar scheme permits you to "digitally sign" a message, that is, to generate a non-forgeable tag that demonstrates that the message really did come from you. This leads to the possibility of so-called "privacy-enhanced" systems—electronic mail systems in which messages are both secure (can be read only by the intended recipient) and authenticated (all messages are signed by the sender in a way that cannot be forged).

This may all seem remarkable (and it is!). Yet there is a public-key encryption technique called RSA that is extremely simple to implement. Even though RSA is easy to implement, you are legally prohibited from developing a communication system that incorporates this method, and letting others use it, unless you obtain a license from the company that has patent rights to the RSA method. Included in this problem set is a copy of the first page of the patent, so you can see what patents look like. The entire patent is about 20 pages long.

This problem set asks you to consider a hypothetical situation in which an MIT undergraduate infringes the RSA patent, and the consequences that might result. First, we'll present some background on RSA. It is a complex story, raising issues of national security, patents that involve software, and university licensing of technology; and parts of the story are still being played out.

The possibility of a public-key system was (as far as is known[2]) first realized by Whit Diffie and Marty Hellman at Stanford University and described in a paper published in 1976.[3] This paper presented the basic idea of public-key cryptography, along with an implementation based on a mathematical problem known as the "knapsack problem"—writing a large integer as the sum of multiples of a fixed set of integers.

Ron Rivest, Adi Shamir, and Len Adelman, all professors at the MIT Laboratory for Computer Science, learned about Diffie and Hellman's work, and, in the spring of 1977, devised the RSA method as a simple, but effective alternative to the Diffie-Hellman knapsack technique. Not only is RSA easier to implement, but it is more secure. After 15 years of scrutiny and analysis, no one has yet found a way to break RSA.

In the fall of 1977, Stanford, based on Diffie and Hellman's work, applied for a patent on communication systems that use public-key cryptography; MIT applied for a patent on communication systems that use RSA, based on the work of Rivest, Shamir, and Adelman.

While the patents were pending, the National Security Agency became concerned about "civilian" scientists becoming increasingly interested in cryptography. Many people think this is because the NSA was afraid that this would make it easy for people to send messages that can not be decoded by the NSA.[4] For whatever the reason, the NSA put pressure on MIT to classify the RSA work, or at least to give NSA the right to review and censor further papers in this area. MIT refused to comply with either request.[5]

The Diffie-Hellman patent was finally granted in 1980. The RSA patent was granted in 1983 and thus is in force until the year 2000. Rivest, Shamir, Adelman and others, formed a company called RSA Data Security, and MIT licensed the RSA patent to this company.[6] The MIT license is an *exclusive license*, which means that MIT has agreed not to license this to anyone else; so anyone

---

[2]The reason for the parenthetical remark is that the National Security Agency has (unofficially) implied that these techniques were known within the intelligence community, long before Diffie and Hellman and RSA, but no evidence to support this has ever been produced .

[3]W. Diffie and M. Hellman, "New directions in cryptography" *IEEE Transactions on Information Theory*, IT-22:6, 1976, pp. 644–654.

[4]Even today, it is *illegal* to send cryptographic software outside the country. For the purposes of the law, cryptographic software is classified as a *weapon* and is controlled by the State Department under the International Traffic in Arms Regulations (ITAR). Exporters of cryptographic software must register as munitions dealers and obtain special licenses to export this software. So far, the government has refused to grant export licenses for RSA or other "strong" cryptographic methods.

[5]NSA has since instituted a procedure whereby authors can voluntarily submit papers for review by NSA for possible national security implications.

[6]The actual details of the original license and its various revisions are confidential. Part of the terms involve royalty payments and stock. MIT now owns about 1% of stock in RSA Data Security.

wanting to use the RSA method before the year 2000 must deal with this one company.[7]

At about the same time, Stanford licensed the Diffie-Hellman patent, also exclusively, to a company called Cylink. Cylink/Stanford accused RSA/MIT of patent infringement, saying that the RSA method, as an implementation of a public-key cryptosystem, infringed the Diffie-Hellman patent, which they claimed covered essentially any public-key system. This raised the possibility of a court fight, and led to confusion about whether people wanting to use the RSA method would have to deal with RSA or Cylink or both. In order to solve these problems, the parties agreed to join forces, and in 1990 they formed a third company called Public Key Partners (PKP). PKP would hold both the Stanford and the MIT patents for RSA Data Security and Cylink, which would continue to work on their individual products. MIT and Stanford then renegotiated their licenses exclusively to PKP.

Over the past few years, RSA Data Security has developed a line of cryptographic products that use the RSA method, and has marketed these to a wide range of companies, both as stand-alone products and as packages that companies can include in their own systems.[8] They have also been promoting a standard for digital certificates that would support a national system of privacy-enhanced electronic mail over the Internet, with certificates authorized by RSA or its licensees.[9]

People who wish to use the RSA method without actually buying the software developed by RSA Data Security must get a license from Public Key Partners, which holds the RSA patent.[10] PKP has been threatening people who develop "unauthorized" implementations of RSA or similar algorithms for public use, either commercially or noncommercially.

Here, for example, is a letter sent in 1991 via electronic mail to Mark Riordan, a staff member at Michigan State University. This was followed by a letter to MSU. Riordan had announced the availability of a public domain "privacy enhanced mail system" for use in sending messages over the Internet:

```
                              May 16, 1991
Dear Mr. Riordan,

We refer to your posting to pem-dev@tis.com of May 16, 1991:

> Announcing the initial release of "rpem", a mostly public domain
> Privacy Enhanced Mail program incorporating a public key encryption system

>  The public key encryption algorithm used in rpem is Rabin's:
```

[7] The only exception is that RSA must be made available at no cost "for government purposes." This was a condition of the National Science Foundation contract to MIT, which funded the RSA work.

[8] For example, Lotus *Notes* uses encryption software supplied by RSA Data Security.

[9] You might wonder why anyone needs a standard here, since the whole idea of public-key encryption is that you can publish your personal key and then people can send you messages as they please. But suppose that the public keys are stored in some data file; someone might substitute their public key for yours and intercept mail intended for you. To prevent this, you use a *certificate*—a kind of key that is "digitally signed" by some authorized person in a way that cannot be forged. (This kind of non-forgeable "digital signature" is also implemented using the RSA algorithm.) The idea of combining a key and a digital signature to form a certificate was invented by Loren Kohnfelder in an MIT Bachelor's Thesis in 1978.

[10] PKP and RSA are technically separate companies, but they are closely connected—the president of RSA is also the president of PKP.

```
>  ciphertext = plaintext^2 mod pq (p, q are primes). The public
>  component of the key is pq, and the private component is p and q.
>  Rabin's algorithm is probably slower (on decryption) and less
>  aesthetically pleasing than RSA, for instance, but it's in the
>  public domain.  Also, unlike RSA, breaking Rabin's scheme is provably
>  as hard as factoring a product of two primes.

        The Massachusetts Institute of Technology and the Board of
Trustees of the Leland Stanford Junior University have granted Public
Key Partners exclusive sublicensing rights to the following patents
registered in the United States, and all of their corresponding
foreign patents:

        Cryptographic Apparatus and Method
        ("Diffie-Hellman") ......................... No. 4,200,770

        Public Key Cryptographic Apparatus
        and Method ("Hellman-Merkle") .............. No. 4,218,582

        Cryptographic Communications System and
        Method ("RSA") ............................. No. 4,405,829

        Exponential Cryptographic Apparatus
        and Method ("Hellman-Pohlig") .............. No. 4,424,414

        These patents cover most known methods of practicing the art
of public-key cryptography, including the system commonly known as
"Rabin," which is NOT, contrary to your claim, public domain, and is
covered by at least two of the patents listed above.

        WE HEREBY PLACE YOU AND ALL USERS OF YOUR IMPLEMENTATION OF
PUBLIC KEY, ON NOTICE THAT THEY ARE INFRINGING ON THESE PATENTS AND WE
RESERVE ALL OF OUR RIGHTS AND REMEDIES AT LAW.

                                Yours,
                                Public Key Partners
                                Robert B. Fougner, Esq.
                                Director of Licensing
```

This is the kind of letter that many companies send to people they claim are infringing their patents. In general, the US patent system requires the patent holder to pursue patent infringers, or else risk losing the patent rights.[11]

Two years ago, RSA Data Security developed an encryption package that can be licensed for free.

[11] Riordan's announcement (quoted in the letter) about the Rabin method being provably as hard as factoring is slightly misleading, because the method he uses is actually not as secure as RSA. The so-called *chosen-message attack*, whereby you trick your opponent into sending messages of *your* choice and then analyze the encoded messages, can be used to break the Rabin method, but (so far as anyone knows) not RSA. For many purposes, though, Riordan's system might provide adequate security. Notice PKP's claim that the RSA patent covers the Rabin method of public-key encryption, despite this qualitative difference from RSA. PKP claims that the MIT patent covers any public-key system that uses exponentiation.

Riordan rewrote his in his privacy enhanced mail system and obtained permission from PKP to use this new encryption package. PKP however, still claims infringement against people who develop independent versions of RSA encryption without getting a license.

Part 3: A hypothetical scenario

Now that you have some background on RSA, we'd like you to consider the following imaginary story. The assignment for this problem set will ask you to place yourself in the role of one of the characters.

Alyssa P. Hacker is an MIT senior who took 6.001 in the spring of her freshman year, where she learned about RSA. She was so impressed with the idea that she designed a "secure message system" that uses RSA to permit people to send encrypted email.

Over summer vacation after freshman year, Alyssa got a job helping her old high school set up its computer lab. The lab contained not only computers, but also modems that enable students and teachers to send messages, files, and programs, both to other schools in the state, and via national networks and bulletin boards. Alyssa included her system as one of the programs for people in the school to use in sharing files and messages, both within the school and with other schools.

Alyssa also uploaded her programs to a national bulletin board, together with a note saying that she and her high school were supplying it as a free program that can be freely redistributed. Over the past three years, these programs have come to be used widely over various networks, both educational networks and others. Her high school guesses that about 100,000 students, teachers, and others around the country are using the program in sending mail and files, but they have really no way to know, since the program is widely redistributed.

In September of this year, Alyssa got a call from the Superintendent of Schools for her high-school's district. The school system has just received a letter from Public Key Partners, claiming that the widespread use of Alyssa's software is interfering with RSA Data Security's product line, and pointing out that Alyssa's system is infringing PKP's patent. The superintendent is very worried. There is no way for the school to keep track of who is using Alyssa's program, and there is absolutely no money in the school budget to pay for licensing fees.

Alyssa is extremely upset. As far as she is concerned, she merely implemented a simple program that she learned about in an MIT course, and let people use it for free. She furious is at her 6.001 instructor, who never mentioned anything in class about RSA being patented. When she goes to complain to him, she finds that he left MIT two years ago. She contacts the Dean of Students, who refers her to the MIT Technology Licensing Office, which is responsible for negotiating all MIT licenses. MIT has the continuing obligation to demonstrate that its licensing policies for government-funded research are in the best public interest, and so the TLO is anxious to defuse any potential conflict arising from an MIT license. They arrange a meeting, at which Alyssa, a member of the TLO, a representative from PKP, and the school superintendent will get together to try to resolve the problem. The meeting will take place on Wednesday, 3 May.

Part 4: Preparing for Friday recitation

In Friday recitation, April 19, you will be divided into groups of four and asked to "role play" the meeting. Each of you will be take one of the roles: Alyssa (or "Alex," if you prefer a male role), the school superintendent, the representative from MIT's TLO, and the representative from PKP. Then you will be asked to reenact the meeting again, this time playing a different role. Thus, you will have to think through the situation from more than one perspective.

Each of the two role-paying sessions should last for about 20 minutes. You should try to remain in character and not act in ways that you would not act if you were actually in that position. The goal of your group of four is to come to some sort of resolution, if possible. The resolution may be a mutually agreeable one, but this might not be possible, and your meeting might end with the threat of a lawsuit among various of the parties.[12]

Your two roles for Friday will be assigned beforehand (in recitation on Friday, April 12), so that you will have time to think through your positions and to prepare the points you want to make. Think about the situation from the point of view of all four participants in the meeting. Why is this a tough situation? What might each participant want from the meeting? What kinds of solutions would he or she find unacceptable? How would you initially approach the other parties? What might you expect them to want or do? How would you respond to each of the approaches that they might take?

Part 5: To write up and hand in on Friday, April 19

Prepare and hand in a short (one page) outline of some possible outcomes you see emerging from the meeting, and briefly state what the favorable and unfavorable aspects of each might be, from the points of view of your assigned roles.

Part 5: To write up and hand in with Problem Set 9, April 24 or 26

Write a brief summary of your two enacted meetings (four pages typewritten, at most). List which students were in your group and who played which role. For each meeting, list the two or three most important points made by each participant. Describe what the resolution was. Also say whether you (in your role) are satisfied or not with the outcome of the meeting. If not, what might you have done differently either at or before the meeting in order to produce a different outcome?

Also, please write one or two sentences saying what you think of including this kind of material in 6.001 and what you think of this assignment. We'll appreciate any suggestions you can make about how this assignment could be improved.[13]

[12]Bear in mind that a lawsuit is not an easy way out. Preparing for a suit like this would cost at least $50,000 in legal fees, and actually going to court could cost over $1 million.

[13]Developing a new format for a 6.001 problem set is a lot of work, and this one, especially, required a lot of background research. I want to thank John Preston, Ron Rivest, and Caroline Whitbeck for their advice and their information.—H.A.