Cryptology
Assignment

Due Friday, April 20, at 10:15.

Note that this is an actual assignment which must be approved in order to take the exam in June. You may not work with others on the assignment, except where it is clearly required. It must be turned in by 10:15 on Friday, April 20. If it is late, it will not be accepted. There will be no chance to redo the assignment if it is not acceptable the first time.

The first part of this assignment is try using PGP. If you are using LINUX, you can get a User's Guide to PGP from

http://www.pgpi.org/doc/guide/6.5/en/cmdline/.

If you are using Windows, you can get a description from

http://www.pgpi.org/doc/guide/6.5/en/win/.

There is further help if you are using Windows from an assignment for the IT course: http://www.imada.sdu.dk/ joan/IT/notes.html (Week 50). This probably makes it much easier to use the Windows system the faculty maintains for all natural science students.

1. Create a pair of keys, public and private. Use the defaults recommended.

2. Try encrypting and decrypting a file.

3. Send your public key to the other two students in the class. Sign the message you send to the others.

4. Read the messages you get from the other students. Check the signatures. Put the public keys on your public keyring. Verify the public keys you received by checking the fingerprints and comparing with the fingerprints on the other students' actual public key.

5. Encrypt some mail with each of the other students' public keys and send it.

6. Decrypt the mail you get from the other students.

7. Write a report describing what you did and what PGP did. What key is used for what? What is a keyring? What is a fingerprint? What is the purpose of a fingerprint? Your report should be about three pages long. (Write the report alone.)

8. Use *help* in Maple to find out about the function *ithprime*. Experiment to find out approximately how big a prime it can find. How large is it? Multiply two of the large primes it finds together, and try to factor the result, using the function *ifactor*. Notice how quickly the factors are found for these small numbers. (Large numbers are clearly necessary for security.)

9. Finding larger primes.

In order to find good prime factors p and q for use in RSA, one can choose random numbers of the required length and check each one for primality until finding a prime.

Maple contains a function *isprime* which will test for primality. Try it on some some small numbers, such as 3, 4, 7, 10. Maple has another function *rand* which returns a random 12-digit number. Try checking three random 12-digit numbers for primality. Try factoring them too? How long do these operations take? Try producing random 12-digit numbers until you find a prime. You can use a *while loop* You want to continue creating new random numbers until you get a prime, so you can type `while (not isprime(x)) do`, ignore the warning, and type `x:=rand();` on one line and `end do;` on the next. How many different values were chosen before a prime was found? (I got 32, but you could get another number.) Now create a second prime called y (remember that y will need some value before your start your *while loop*). Multiply x and y together and try factoring the result. This should also go relatively quickly.

10. Even larger numbers. Use `rand()` to find even larger primes, multiply them together and try factoring them. How large do the primes have to get before it takes a long time to factor them? Can you find primes which are

much longer? Investigate this to see for different length numbers how long it takes to find a prime and how long it takes to factor a number of that length. Does it make any difference if the numbers to be factored have small prime factors?

Problems

To be discussed Wednesday, April 4, 2001.

1a. In one of the homework problems this semester, you showed that using RSA in the following way was insecure: Use a large modulus n for which it is unlikely that anyone can factor n in a reasonable amount of time. Represent each alphabetic character (from English) as an integer between 0 and 25. Encrypt each character of a message separately using the number which represents it.

Suppose instead of using a large composite n and RSA, a large prime p and the El-Gamal public-key cryptosystem is used. Is this also insecure? Why or why not?

1b. Are there any advantages or disadvantages of using El Gamal this way, other than those related to security that you discussed above?

2. Do problems 5.3, 5.7, 8.3 and 8.5 in the textbook. For 8.5, you do not need to know the MTI protocol, onl the MTI problem.