ACCESS 2005

Final project - Week 1 - Due Friday, June 24

Part I Set up an RSA public-key cryptography system for the 7 ACCESS groups. Follow the steps below carefully!!!

(1) Each team should choose two primes between 10^30 and 10^31, so that the modulus formed by their product is greater than 10^60 - and thus will have at least 61 digits. This means that when you send people messages you can use up to 60 digits in each packet , **before** encoding, staying safely the residue range for their modulus. (After you encode a packet it will quite likely have 61 or 62 digits, but it will still be less than the receiver's modulus because of how encoding works.) Real RSA systems use much larger primes, but I have chosen these lengths so that each packet of information fits onto a single text line.

(2) After picking your primes find a suitable encryption power e, then use e to compute your secret decryption power d. Make sure e and d are multiplicative inverses mod N2, and check that you can encrypt and decrypt your own messages.

(3) Send a plain text email to the ACCESS list identifying your group, your public key information, and the email address(es) people should use to contact you during this game. Make sure your encryption power e and modulus N are text numbers and NOT a picture image. (You can convert a number which is a Maple output picture into text by copying it as "Maple text".)

(4) Nick, Meagan and Sarah will try to be troubleshooters, so send each of us emails which identify your group, email contact address, as well as ALL your public and private information: p,q,N2,e,d. When I (Nick) get this data I'll check to see it's all O.K., and then I will create a master list of everyone's public keys which I can link to our home page. I hope to complete this task Friday morning. Send Nick's email to his UNID account, njk4@utah.edu. Sarah and Meagan will tell us which accounts they want you to send to.

(5a) Create a cool secret message and signature, and convert your letters to numbers using the table on page 9 of Davis' notes. Let us agree that your message will end with a sentence ender (period, exclamation mark or question mark) and a space, so that after conversion it ends in 7010, 6310, or 8610. Let us agree that no secret message is longer than 88 letter-punctuation characters long. If you have a longer message send part of it as plain text and only encrypt the best parts.
(5b) We are using the secure signature feature, so decrypt you signature using your own (secret) decryption power. This will create a long sequence of gibberish, which you glue onto your plaintext message.
(5c) NOW, break the numerical message with the gibberish appended, into packets. Make each packet at most 60 digits long. You will probably find that your gibberish fills all of your last packet and part of the preceding one.
(5d) Being at most 60 digits long, your packets are all less than everybody elses moduli "N", so after you encrypt and they decrypt each packet you send, they should recover your original message chunk. (If you encrypt a message bigger than the receiver's modulus, a common ACCESS error, then when the receiver decrypts, they will have a number congruent to your message, but in their residue range. Since this is a different number than you sent the result is very bad news.) Encrypt and send your message to two groups: the groups with residue numbers one less than yours and one greater than yours, mod 7. For example, group 3 sends messages to groups 2 and 4; group 7 is congruent to group 0, so sends a message to group 6 and to group 1. Please use the same plain text message and signature for both groups. (Of

course, after you encrypt using the different groups' keys, you will be sending different ciphertexts.) Since your original message had at most 88 letters (=176 digits) and since your decrypted signature is less than 10^62 (<63 digits), you should have at most 239<240 digits before encrypting, so at most 4 packets of length up to 60 digits will suffice.

(6) Use your private key and reverse the encryption process to decode the messages you receive from your two neighbor groups. Using the fact that their message ends in 7010, 6310, or 8610, identify and re-glue together the gibberish which is their decrypted signature, and then use that group's encryption key to read their signature. (This part always gives fits to a few of the ACCESS groups.)

Part II Create a project report. The first section of your project report should be a 2-4 page discussion of the general ideas behind public key cryptography. Explain why its advent was such a revolutionary development. Explain the RSA algorithm. We have given you plenty of source material for this part of your report, but feel free to do more research if you wish. Make sure to cite all references.

In the second section of your report describe the process you went through to set up the ACCESS RSA system. Exhibit your public and private keys, your original plain text message and signature, and the various transformations of your message and signature as you prepared them for transmission to your two target groups. Exhibit the encoded messages you received, explain how you decoded them, and exhibit the final results. Make sure all numerical representations of your messages are numbers and not pictures, so that I will have an easy time checking your work. Explain well.

Although different subjects and Professors may have different standards for how papers should be formatted, there are certain common elements. Reports should begin with a cover page or title area, containing the title, the authors, and the date. This should be followed with an introduction which summarizes the report's contents. The body of the project report may be split into sections, and a conclusion section may be appropriate. Include references, URL's, and footnotes if you have them. The RSA paper by Rivest-Shamir-Adelman is an excellent example of how to write a paper in mathematics.

Paper submission: Please submit your paper to me (Nick) as an attachment in an email. Send the email to my Math address, korevaar@math.utah.edu. Your document should either be in Microsoft Word or Word Perfect format. This project is due by Friday June 24, at 5 p.m. Help each other! If you think a group sent you a defective message, contact them and explain what isn't working. If you get stuck see if Meagan, Sarah or I can help. (I'll be out of town from Friday afternoon until Monday night, but am
available starting Tuesday morning.) I will plan to be available from 3-5 on Wednesday afternoon next week, in the large Marriott computer room, in case any groups are stumped. Please send me an email in that case, so that I know I will be needed. Have fun!!