ACCESS 2006

Group Project - Week 1

Due Thursday, June 22, before midnight

Part I Set up an RSA public-key cryptography system for the 8 ACCESS groups. Follow the steps below carefully!!!

(1) Each team should choose two primes p,q between 10^30 and 10^31, so that the modulus N=pq 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 number packet , **before** encoding with their public key. This forces each packet number to be safely in the residue range for the recipient's modulus before you encrypt it, so that when the recipient decrypts it, they will recover your original message. (After you encode a packet it will quite likely have 61 or 62 digits, but it will still be in the recipient's residue range because of how the encryption algorithm works.) Real RSA systems use much larger primes, but I have chosen these lengths so that each digit packet fits onto a single text line.

(2) After picking your primes find a suitable encryption power e. Use e and the auxillary modulus to compute your secret decryption power d. Make sure e and d are multiplicative inverses mod N2, check that you can successfully encrypt and decrypt messages using your public and private information, and verify that N really is bigger than 10^60, before proceding to step (3) - This double-checking should prevent errors which have occurred in previous ACCESS groups, and which have led to strings of emails with different attempts at a public key, all from the same group. Now would also be a good time for you to save and email all of your public and private key data to all three group members, so that you can recover if you accidentally forget to save work later on. (Convert a number which is a blue Maple output picture into text by copying it as "Maple text". You want to do this, since you won't be able to turn a picture of numbers back into text numbers later on unless you do it by hand.)

(3) Send a plain text email to the ACCESS list identifying your group by number, with your three names, with contact email address(es) for the group, and with your public key information. Again, make sure your encryption power e and modulus N are text numbers and NOT a picture image, so that the recipients can use them.

(4) Nick, Meagan and Erin will try to be troubleshooters. Send Nick ALL your public and private information: group number, email contact info, names, p,q,N2,e,d. When I (Nick) get this data I'll triple-check to see it's all O.K. (and let you know if it is or isn't), 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, but certainly by the afternoon. Send Nick's email to his Umail account, njk4@utah.edu.

(5a) Create a favorite secret message and signature, and convert your letters to numbers using the table on page 9 of Davis' notes. Let us agree that your plain text 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 plain text message is longer than 88 letter-punctuation characters long, so that after table conversion the corresponding list of numbers is at most 176 digits 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 your signature using your own (secret) decryption power. This will create a long sequence of digits (a number less than your groups' modulus but probably with 61 or 62 digits). Append this sequence onto the digits from the converted plaintext message, to create a single long string of numbers.

(5c) NOW, break the single long string (your converted message followed by the digits of your decrypted converted signature), into packets. Make each packet at most 60 digits long. (You will probably find that your signature gibberish fills all of your last packet and a few spaces in the preceding one.) Being at most 60 digits long, your packets are all less than everybody elses moduli "N", and so in their residue range. Thus, after your recipients decrypt each packet you sent and glue the decryptions back together, they should recover your single long string. They will know where your message ends and their signature begins by finding one of the message enders, 7010, 6310, 8610, followed by nonsense numbers.

(5d) Send your message packets to two groups: if you are group x then send messages to groups (x+1) and (x-1), mod 8. For example, group 3 sends messages to groups 2 and 4; group 8, also known as group 0, sends messages to groups 7 and 1. Please use the same plain text message and signature for both recipient groups - so you will be working with the same long string from (5b), for both recipient groups. (Of course, after you encrypt using the their different keys, you will be sending two 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 176+63=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 converted plain text message ends in 7010, 6310, or 8610, separate off their decrypted signature and use their encryption key (and Davis' table) to recover what their signature was. (This part of the project always gives fits to a couple of the ACCESS groups.)

Part II Create a project report.

The first section of your project report should be a 3-5 page discussion of public key cryptography. Explain what it is and why its advent was such a revolutionary development. Explain the RSA algorithm and why it works for public transactions. We have given you various references for this part of your report, but we encourage you to also do more independent research. Questions I would enjoy you finding answers to are: When you engage in secure internet transactions (i.e. at any URL starting with https://....) how much of this interaction is typically made using public key cryptography? Is the RSA algorithm universally used for public key cryptography or are other algorithms also being applied? When RSA is being used, what is the typical modulus size? How is "security certificate" authentication related to the RSA secure signature feature, if at all? Is it now possible to not only send documents, but also to make secure phone calls over the internet, using "pretty good privacy"? What strategies do groups like the National Security Agency adopt in their quest to track potential enemies, in order to get around the fact that public key cryptography apparently allows for secure information transmission?

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 key information, 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. Make sure to cite all references. For internet references include a link to the web page, as well as the site title and author. 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, korevaar@math.utah.edu) and Erin (erin@math.utah.edu), as an attachment in an email. Your document should either be in Microsoft Word or Word Perfect format. This project is due by Thursday June 22, before midnight. Help each other! If you think a group sent you a defective message, contact them and explain what isn't working, as a prelude to both sides trying to troubleshoot the problem. If you get stuck see if Meagan, Erin or I can help. I will also plan to be available from 2-4 on Wednesday afternoon next week, in the large Marriott computer room, in case any groups are stumped. Please send me an email if you wish to take advantage of this meeting time, since if no one asks I won't show up. Have fun!!